

BAB 2

LANDASAN TEORI

2.1 Teori-teori Umum

2.1.1 Analisis dan Perancangan Sistem

2.1.1.1 Pengertian Analisis

Menurut Laudon (2003, p394), Analisis sistem adalah memeriksa masalah yang ada dan akan diselesaikan oleh perusahaan dengan menggunakan sistem informasi.

2.1.1.2 Pengertian Perancangan

Menurut Laudon (2003, p394), Perancangan sistem adalah cara bagaimana sebuah sistem dapat memenuhi kebutuhan informasi yang telah ditentukan oleh analisa sistem.

2.1.2 Pengantar *Data Warehouse*

2.1.2.1 Pengertian Data

Data merupakan elemen dasar yang di miliki setiap transaksi, semua hal memiliki data. Tentang definisi data, menurut Inmon (2005, p493), Data adalah sebuah rekaman dari fakta, konsep, ataupun instruksi pada sebuah media penyimpanan untuk komunikasi, pengambilan, maupun pemrosesan dari pengertian

otomatis dan presentasi dari informasi yang dapat di mengerti oleh manusia.

Data menurut Turban et al. (2005, p38), adalah deskripsi dasar tentang sesuatu, kejadian, kegiatan, dan transaksi yang ditangkap, direkam, disimpan, dan diklasifikasikan, namun tidak terorganisir untuk menyampaikan suatu arti khusus.

Dari pengertian diatas, data merupakan bagian dari informasi yang berupa rekaman dari fakta dan haruslah dimengerti dalam bahasa manusia. Namun kata mengerti dalam konteks ini bukanlah dalam artian analisis. Jadi data disini hanya sekedar interpretasi dari data konkrit tanpa dapat dianalisa lebih lanjut, dan tingkat kegunaannya untuk kepentingan manajerial perusahaan tidak terlalu tinggi bila berdiri sendiri.

2.1.2.2 Pengertian *Database*

Menurut Connoly (2005, p14), *database* adalah suatu kumpulan dari data yang terselubung secara logis, dan deskripsi dari data ini, dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi. *Database* bersifat tunggal, memiliki tempat penyimpanan data yang besar di mana dapat digunakan secara bersama-sama oleh banyak Kementerian dan *user*.

Menurut Inmon (2002, p388), *a database is collection of interrelated data stored (often with controlled, limited redundancy)*

according to a schema, yang dapat diartikan bahwa *database* adalah sekumpulan penyimpanan data yang berhubungan (sering dengan pengontrolan, redundansi yang terbatas) yang berdasarkan suatu skema.

Berdasarkan definisi di atas, dapat disimpulkan bahwa *database* merupakan suatu kumpulan data yang saling berhubungan dan terintegrasi, yang di rancang untuk kebutuhan informasi perusahaan.

2.1.2.3 Pemodelan *Entity relationship*

Konsep dasar dari model ER adalah tipe-tipe entitas, yaitu kumpulan dari objek-objek yang bersifat (*property*) yang sama, yang diidentifikasi oleh perusahaan mempunyai eksistensi yang independen. Keberadaannya dapat berupa fisik atau abstrak.

Entity Occurrence, yaitu pengidentifikasian objek yang unik dari sebuah tipe *entity*. Setiap entitas diidentifikasi dan disertakan *property*-nya.

Relationship Occurrence, yaitu hubungan yang diidentifikasi secara unik, yang meliputi keberadaan tipe-tipe entitas yang berpartisipasi.

2.1.2.4 *Multiplicity*

Batasan utama pada *relationship* disebut *multiplicity*, yaitu jumlah (atau *range*) dari kejadian yang mungkin terjadi pada suatu

entitas yang terhubung ke satu kejadian dari entitas lain yang berhubungan melalui suatu *relationship*.

Relationship yang paling umum adalah *binary relationship*.

Macam-macam *binary relationship* yaitu :

- *One-to-one* (1 : 1)
- *One-to many* (1 : *)
- *Many-to-many*(* : *)

Summary untuk *Multiplicity Constrains*

Alternative ways to represent multiplicity constraints	Meaning
0..1	Zero or one entity occurrence
1..1 (or just 1)	Exactly one entity occurrence
0..* (or just *)	Zero or many entity occurrences
1..*	One or many entity occurrences
5..10	Minimum of 5 up to a maximum of 10 entity occurrences
0, 3, 6-8	Zero or three or six, seven, or eight entity occurrences

Gambar 2.1 Macam – Macam *Binary Relationship*.

2.1.2.5 Pengertian *Data Mart*

Data mart merupakan bagian dari *data warehouse* yang mendukung kebutuhan sebuah Kementerian atau fungsi bisnis tertentu. Sebuah *data mart* menyimpan sebagian data kedalam *data warehouse*, biasanya berupa data yang berisi ringkasan yang

biasanya berhubungan dengan sebuah Kementerian atau fungsi bisnis tertentu.

Karakteristik yang membedakan *data mart* dengan *data warehouse* antara lain:

- a. Sebuah *data mart* hanya berfokus kepada kebutuhan dari pengguna yang berhubungan pada suatu departmen atau fungsi bisnis.
- b. *Data mart* berisi data yang lebih sedikit dibanding *Data Warehouse* sehingga lebih mudah dimengerti.
- c. *Data mart* biasanya tidak berisi fungsi oprasional yang mendetail seperti *Data Warehouse*.

Ada beberapa pendekatan untuk membangun *data mart*. Salah satunya adalah dengan membangun *data mart* yang akan berkelanjutan untuk diintegrasikan menjadi sebuah *data warehouse*. Pendekatan lainnya adalah dengan membangun infrastruktur *data warehouse* perusahaan pada saat bersamaan dibangun pula satu *data mart* atau lebih untuk memenuhi kebutuhan bisnis yang berlangsung.

2.1.2.6 Pengertian OLTP

OLTP (*Online Transaction Processing*), menurut Kimball dan Ross (2002, p408) adalah “Deskripsi awal dari setiap aktifitas dan sistem yang berhubungan dengan proses memasukkan data kedalam sebuah *database*”.

Menurut Inmon (2005, p500), OLTP adalah lingkungan pemrosesan transaksi performance tingkat tinggi. OLTP (*Online Transaction Processing*) menggambarkan kebutuhan sistem dalam ruang lingkup operasional dan merupakan proses yang mendukung operasi bisnis sehari-hari.

OLTP dirancang untuk memungkinkan terjadinya pengaksesan secara bersamaan oleh beberapa *user* terhadap sumber data yang sama dan mengatur proses yang diperlukan. Sesuai dengan namanya, OLTP mengizinkan transaksi untuk mengakses langsung ke *database*. Transaksi yang dilakukan termasuk operasi *insert*, *update*, dan *delete*. *Database* OLTP biasanya bersifat relasional dan dalam bentuk normalisasi ketiga, serta yang terpenting, *database* OLTP dibangun untuk mampu menangani banyak transaksi dengan performa tinggi.

Jadi dapat disimpulkan, bahwa OLTP memungkinkan banyak *user* untuk mengakses secara bersamaan terhadap sumber *database* yang sama, dimana *database*-nya bersifat relasional dan sudah ternormalisasi.

2.1.2.7 Pengertian OLAP

OLAP (*Online Analytical Processing*) adalah sebuah pendekatan untuk menjawab multi-dimensional *analytic queries*. Menurut Depak Pareek (2007, p294), OLAP sering dikategorikan

sebagai sebuah *Business Intelligence*, yang juga mencakup *relational reporting* dan *data mining*.

Terminology dari OLAP sendiri dicantumkan dalam OLAP *Council White Paper* (1997) sebagai modifikasi terhadap *terminology database* pada umumnya (OLTP).

2.1.2.8 Pengertian Data Warehouse

Berdasarkan Connolly (2002, p1047), *Data Warehouse* adalah koleksi data yang mempunyai sifat berorientasi pada subjek, terintegrasi, rentang waktu, dan koleksi datanya tidak mengalami perubahan dalam mendukung proses pengambilan keputusan di manajemen.

Sedangkan menurut S. Prabhu dan N. Venatesan (2007, p4), *Data Warehouse* adalah sebuah koleksi dari *decision support technologies*, yang memiliki tujuan untuk mengaktifkan *knowledge worker* (*executive, manager, analyst*) untuk membuat keputusan yang lebih baik dan lebih cepat.

2.1.2.9 Perbandingan Data Operasional dan Data Warehouse

Data operasional adalah kumpulan data yang memiliki sifat *real time*, dimana *database* mencatat seluruh transaksi yang terjadi dan baru saja terjadi. Sehingga sering sekali terjadi perubahan data di dalamnya, baik itu berupa *Insert, Update*, ataupun *Delete* terhadap *record* yang sudah ada sebelumnya.

Sedangkan *Data Warehouse* tidak memiliki sifat yang *real time*, dimana *data warehouse* akan mengambil data secara *periodic* berdasarkan waktu yang sudah ditentukan, sehingga *data warehouse* tidak banyak terjadi transaksi *update*, dan *delete*, karena data yang dimasukan adalah data hasil proses dari operasional data. Oleh karena itu *data warehouse* cocok digunakan sebagai *summary*, dan *archiving* terhadap data-data yang ada.

Tabel 2.1 Perbandingan Data Operasional dan *Data Warehouse*

Data Operasional	<i>Data Warehouse</i>
Bersifat real-time	Tidak bersifat realtime
Sering sekali terjadi perubahan data pada <i>database</i> (sering terjadi <i>insert, update, delete</i>)	Data hanya dimasukan pada periode waktu tertentu
Tidak cocok digunakan sebagai <i>report</i>	Cocok digunakan sebagai <i>report</i> karena bersifat <i>summary</i>
Tidak baik untuk menyimpan data pada waktu yang lama	Baik digunakan untuk melakukan <i>archiving</i> data

2.1.2.10 Keuntungan *Data Warehouse*

Banyak perusahaan dan lembaga-lembaga besar yang menggunakan *data warehouse* sebagai referensi data dimana kemudian mereka mengambil keputusan dari sana, namun selain itu masih banyak keuntungan-keuntungan lain, seperti:

- a. *Data Warehouse* membantu para pekerja (*manager, executive, dan analyst*) untuk pengambilan keputusan yang lebih cepat dan mudah.
- b. *Data Warehouse* memungkinkan *user* untuk memeriksa *history* data dan melakukan analisis terhadap data sehingga dapat mengambil keputusan berdasarkan analisa yang dibuat.
- c. Pengembalian investasi yang tinggi.
- d. Keuntungan yang kompetitif.
- e. Meningkatkan produktifitas para pembuat keputusan korporasi.

Manfaat dari *data warehouse* itu sendiri juga banyak, salah satunya adalah :

- a. Kemampuan untuk mengakses data yang berskala perusahaan.
- b. Kemampuan memiliki data yang konsisten.
- c. Dapat melakukan analisis secara cepat.
- d. Dapat digunakan untuk mencari redudansi usaha di dalam perusahaan.
- e. Menemukan jarak antara pengetahuan bisnis dan proses bisnis.
- f. Meminimalkan biaya administrasi.

- g. Meningkatkan kinerja pegawai perusahaan untuk dapat melakukan tugasnya dengan lebih efektif.

2.1.2.11 Karakteristik *Data Warehouse*

Menurut Connolly (2002, p1047) dan Inmon (2002, p31), *data warehouse* memiliki karakteristik sebagai berikut :

2.1.2.11.1 *Subject Oriented*

Data Warehouse di desain untuk menganalisa data berdasarkan subjek-subjek tertentu dalam organisasi, bukan berorientasi kepada proses atau fungsi aplikasi tertentu. Sebagai contoh, pada KESDM fungsi aplikasinya terdiri dalam ekspor dan impor, sedangkan dalam penerapannya di dalam *data warehouse* data akan di kategorikan kedalam subjek seperti minyak bumi, gas dan listrik.

2.1.2.11.2 *Integrated*

Data di dalam *data warehouse* bersifat *ter-integrated* karena *data warehouse* harus menyimpan data-data yang berasal dari sumber-sumber yang terpisah dan tidak konsisten satu dengan lain kedalam suatu format yang konsisten dan saling terintegrasi satu dengan yang lainnya. Dengan demikian data tidak bisa dipecah-pecah karena data

yang ada merupakan satu kesatuan yang menunjang keseluruhan konsep *data warehouse* itu sendiri.

2.1.2.11.3 *Non-Volatile*

Non-Volatile merupakan keadaan dimana data-data yang terdapat di dalam *data warehouse* merupakan data-data yang tidak dapat di ubah isinya (*read-only data*). Hal ini menetapkan bahwa data tersebut sudah konkrit dan tidak dapat di utak-atik lagi.

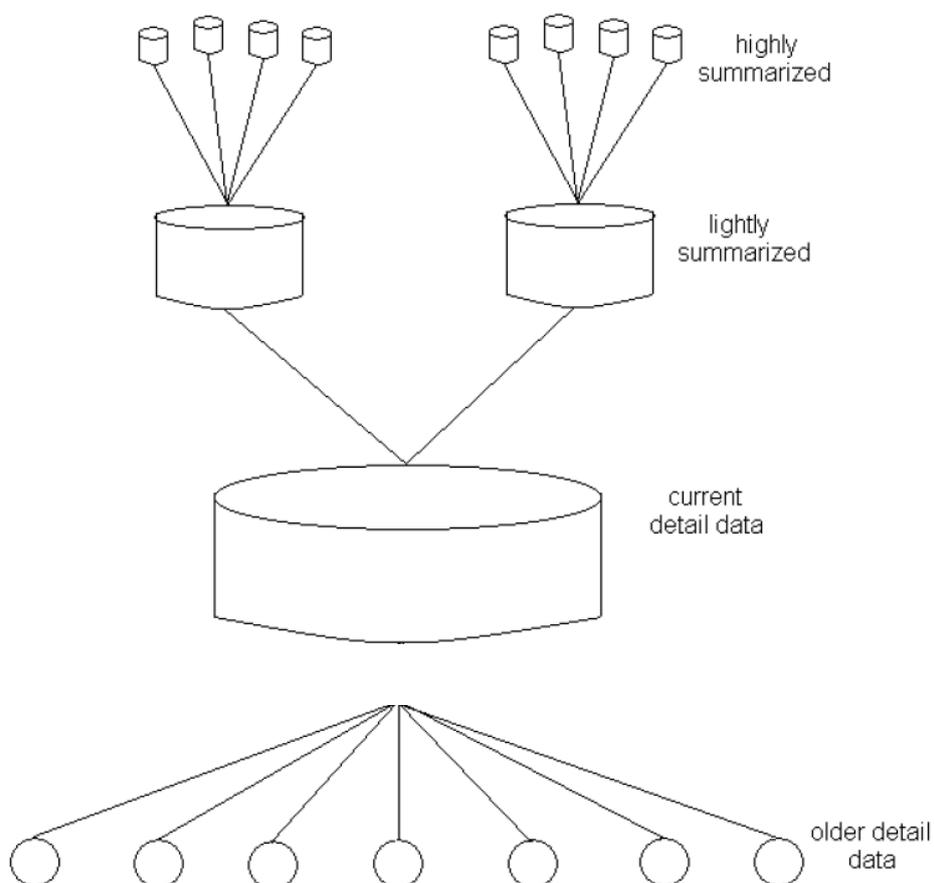
2.1.2.11.4 *Time-Variant*

Data pada *data warehouse* bersifat *time-variant* karena data-data tersebut memiliki keterangan waktu di dalamnya yang akan membantu mengkategorikan data-data tersebut sehingga ketika kita ingin mendapatkan hasil *summary* dari *data warehouse* akan terjadi dengan mudah dan cepat.

2.1.2.12 Struktur *Data Warehouse*

Seperti yang kita lihat pada arsitektur *data warehouse*, ada beberapa struktur yang spesifik terdapat pada bagian *warehouse manager*. Bagian tersebut merupakan struktur *data warehouse*. Menurut Poe, Vidette, *Data Warehouse* memiliki struktur yang spesifik dan mempunyai perbedaan dalam tingkatan

detail data dan umur data. Komponen dari struktur *data warehouse* adalah :



Gambar 2.2 Struktur *Data Warehouse*

2.1.2.12.1

Current Detailed Data

Current detail data merupakan data detail yang aktif saat ini, mencerminkan keadaan yang sedang berjalan dan merupakan level terendah dalam *data warehouse*. Di dalam area ini *warehouse* menyimpan seluruh detail data yang terdapat pada skema basis data. Jumlah data sangat

besar sehingga memerlukan *storage* yang besar pula dan dapat di akses secara cepat. Dampak negatif yang ditimbulkan adalah kerumitan untuk mengatur data menjadi meningkat dan biaya yang diperlukan menjadi mahal. Berikut ini beberapa alasan mengapa *current detail data* menjadi perhatian utama :

1. Menggambarkan kejadian yang baru terjadi dan selalu menjadi perhatian utama.
2. Sangat banyak jumlahnya dan disimpan pada tingkat penyimpanan terendah.
3. Hampir selalu disimpan dalam *storage* karena cepat di akses tetapi mahal dan kompleks dalam pengaturannya.
4. Bisa digunakan dalam membuat rekapitulasi sehingga *current detail data* harus akurat.

2.1.2.12.2 *Old Detailed Data*

Data ini merupakan data historis dari *current detail data*, dapat berupa hasil cadangan atau *archive* data yang disimpan dalam *storage* terpisah. Karena bersifat *back-up*(cadangan), maka biasanya data disimpan dalam *storage* alternatif seperti *tape-desk*.

Data ini biasanya memiliki tingkat frekuensi akses yang rendah. Penyusunan file atau *directory* dari data ini di

susun berdasarkan umur dari data yang bertujuan mempermudah untuk pencarian atau pengaksesan kembali.

2.1.2.12.3 *Lightly Summarized Data*

Data ini merupakan ringkasan atau rangkuman dari *current detail data*. Data ini di rangkum berdasar periode atau dimensi lainnya sesuai dengan kebutuhan.

Ringkasan dari *current detail data* belum bersifat total *summary*. Data-data ini memiliki detail tingkatan yang lebih tinggi dan mendukung kebutuhan *data warehouse* pada tingkat Kementerian. Tingkatan data ini disebut juga dengan *data mart*. Akses terhadap data jenis ini banyak digunakan untuk *view* suatu kondisi yang sedang atau sudah berjalan.

2.1.2.12.4 *Highly Summarized Data*

Data ini merupakan tingkat lanjutan dari *lightly summarized data*, merupakan hasil ringkasan yang bersifat totalitas, dapat di akses misalnya untuk melakukan analisis perbandingan data berdasarkan urutan waktu tertentu dan analisis menggunakan data multidimensi.

2.1.2.12.5 **Metadata**

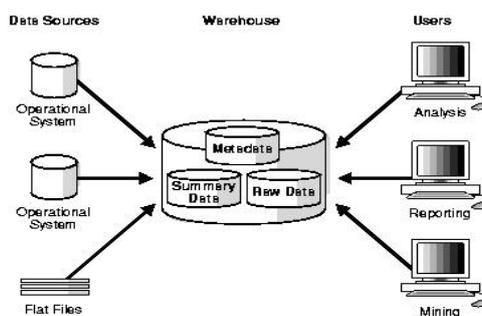
Metadata bukan merupakan data hasil kegiatan seperti keempat jenis data diatas. Menurut Poe, metadata adalah ‘data tentang data’ dan menyediakan informasi

tentang struktur data dan hubungan antara struktur data di dalam atau antara *storage* (tempat penyimpanan data).

Metadata berisikan data yang menyimpan proses perpindahan data meliputi *database structure*, *contents*, detail data dan *summary data*, *matrics*, *versioning*, *aging criteria*, *versioning*, *Transformation criteria*. Metadata khusus dan memegang peranan yang sangat penting dalam *data warehouse*. Metadata sendiri mengandung:

1. Struktur data, yaitu sebuah direktori yang membantu *user* untuk melakukan analisis *decission support* Sistem dalam pencarian letak/lokasi dalam *data warehouse*.
2. Algoritma, digunakan untuk *summary data*. Metadata sendiri merupakan panduan untuk algoritma dalam melakukan pemrosesan *summary data* antara *current detail data* dengan *lightly summarized data* dan antara *lightly summarized data* dengan *hightly summarized data*.
3. *Mapping*, sebagai panduan pemetaan (*mapping*) data pada saat data di *transform*/diubah dari lingkup operasional menjadi lingkup *Data Warehouse*.

2.1.2.13 Arsitektur *Data Warehouse*



Gambar 2.3 Arsitektur *Data Warehouse*

Menurut Connolly dan Begg (2002, p1056-p1161), komponen *data warehouse* yang terdapat pada arsitektur *data warehouse*, adalah sebagai berikut:

2.1.2.13.1 Operasional Data

Sumber *data warehouse* diperoleh dari:

- a. *Mainframe* data operasional menangani *database* jaringan dan hirarki generasi pertama.
- b. Data Kementerian berada di sistem penyimpanan file Kementerianal seperti CSV, Excel dan *relational DMBS* seperti MySQL, MSSQL, PostgreSQL.

- c. Data pribadi yang tersimpan dalam *workstation* dan *server* pribadi.
- d. Sistem eksternal seperti *internet*, *database* komersial, atau *database* yang berhubungan dengan *supplier* dan *consumer*.

2.1.2.13.2 Operasional Data Store (ODS)

Operasional *Data Store (ODS)* adalah tempat penyimpanan sementara dari data operasional saat ini yang terintegrasi yang digunakan untuk analisis. ODS memiliki sumber data dan struktur yang sama seperti *data warehouse*, bertindak secara sederhana sebagai tempat penampungan sementara sebelum data di pindahkan ke *data warehouse*.

Membangun ODS merupakan tahap yang berguna dalam membangun *data warehouse*, karena sebuah ODS dapat menyuplai data yang sudah di ekstrak dari sistem sumber dan dibersihkan. Integrasi dan restrukturisasi data untuk *data warehouse* menjadi lebih sederhana.

2.1.2.13.3 Load Manager

Load manager melakukan semua operasi yang berhubungan dengan *extract* dan *load* data ke dalam *data warehouse*. Data di *extract* dari sumber-sumber data atau pada umumnya dari penyimpanan data operasional. Operasi yang

dilakukan *load manager* dapat berupa *transformasi* data yang sederhana untuk mempersiapkan data tersebut agar dapat dimasukkan kedalam *data warehouse*.

2.1.2.13.4 ***Warehouse Manager***

Warehouse Manager melakukan semua operasi yang berhubungan dengan *management* data dalam *data warehouse*. Operasi yang dilakukan oleh *warehouse manager* meliputi :

- a. Analisis data untuk memastikan konsistensi.
- b. Perubahan bentuk dan penggabungan data sumber dari gudang penyimpanan sementara ke dalam tabel *Data Warehouse*.
- c. Membuat indeks dan mengacu pada tabel dasar.
- d. Pembuatan denormalisasi.
- e. Pembuatan agrerasi.
- f. Melakukan *back-up* dan *archive/backup* data.

2.1.2.13.5 ***Query Manager***

Query Manager melakukan semua operasi yang berhubungan dengan pengaturan *user query*. Komponen ini dibangun menggunakan *vendor end-user data access tools*,

data warehouse monitoring tools, fasilitas basis data, dan *custom build-in program*. Kompleksitas *queries manager* ditentukan oleh fasilitas yang disediakan oleh *end-user access tools* dan *database*. Operasi yang dilakukan komponen ini berupa pengarahannya *query* pada tabel-tabel yang tepat dan penjadwalan eksekusi *query*.

2.1.2.13.6 ***Detailed Data***

Detailed data menyimpan semua data detail di dalam skema *database*. Data detail terbagi 2 yaitu :

a. Current detail data

Data ini berasal langsung dari operasional *database*, dan selalu mengacu pada data perusahaan sekarang. *Current detail data* di atur sepanjang sisi-sisi subyek seperti data profil pelanggan , data aktifitas pelanggan, data *sales*, data demografis, dan lain-lain.

b. Old detail data

Data ini menampilkan *current detail data* yang berumur atau histori dari subyek area. Data ini yang dipakai untuk menganalisis *trend* yang akan dihasilkan.

2.1.2.13.7 ***Lightly and Highly Summarized Data***

Lighly and Highly Summarized Data menyimpan semua data yang diringkas oleh *warehouse manager*. Data perlu di ringkas untuk mempercepat performa *query*. Ringkasan data terus di perbaharui seiring dengan adanya data baru yang masuk ke dalam *data warehouse*.

2.1.2.13.8 ***Archive/Backup Data***

Archive/backup data menyimpan data detail dan ringkasan data dengan tujuan untuk menyimpan dan *backup* data. Walaupun ringkasan data dibangun dari detail data, akan memungkinkan untuk membuat cadangan ringkasan data secara *online* jika data ini ditunjukkan melebihi penyimpanan waktu untuk detail data. Data ditransfer ke arsip penyimpanan seperti *magnetic tape* atau *optical disk*.

2.1.2.13.9 **Metadata**

Metadata menyimpan semua definisi metadata yang digunakan oleh semua proses di dalam *data warehouse*. Tujuan metadata adalah sebagai berikut :

- a. Proses *Extract* dan *Load* - metadata digunakan untuk memetakan sumber data ke dalam pandangan umum data dalam *data warehouse*.

- b. Proses manajemen *data warehouse* – metadata digunakan untuk mengotomatisasi pembuatan tabel ringkasan.
- c. Sebagai proses manajemen *query* – metadata digunakan untuk mengarahkan suatu *query* dengan sumber data yang tepat.

2.1.2.13.10 *End-user Access Tools*

Tujuan utama *data warehouse* adalah menyediakan informasi bagi pengguna untuk pembuatan keputusan yang strategis dalam berbisnis. Para pengguna ini berinteraksi dengan *data warehouse* menggunakan peralatan akses *end-users*. Berdasarkan kegunaan *data warehouse*, ada 5 kategori *end-users access tools*, yaitu:

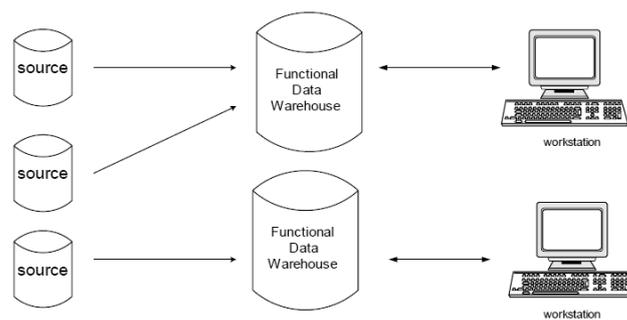
- a. *Reporting and query tool*
- b. *Application development tool*
- c. *Executive information sistem (EIS) tool*
- d. *Online analytical processing (OLAP) tool*
- e. *Data mining tool*

2.1.2.14 Bentuk - Bentuk *Data Warehouse*

2.1.2.14.1 *Data Warehouse Fungsional (Functional Data Warehouse)*

Data Warehouse di buat lebih dari satu dan dikelompokkan berdasarkan fungsi-fungsi pada perusahaan, misalnya seperti fungsi keuangan, marketing, kepegawaian atau personalia, dll.

Bentuk ini memberikan keuntungan berupa sistem yang mudah di bangun dengan biaya yang *relative* murah, sedangkan kerugiannya adalah hilangnya konsistensi dan integritas data akibat terbatasnya kemampuan pengumpulan data bagi para *user*.

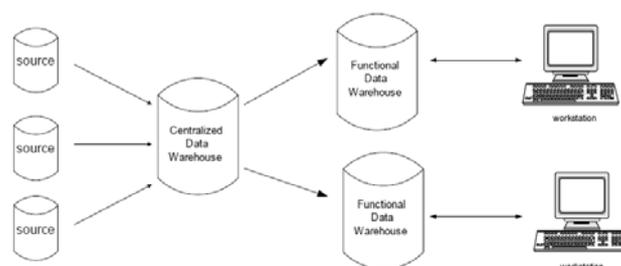


Gambar 2.4 *Data Warehouse* Fungsional

2.1.2.14.2 *Data Warehouse* Terpusat (*Centralized Data Warehouse*)

Data Warehouse di buat pada sebuah *enterprise* data model yang konsisten dan berada pada suatu lokasi *physical* yang tetap, untuk menjamin konsistensi data yang ada ketika dilakukan proses ETL pada *database* ke *data warehouse*. Dengan data yang konsisten maka akan

mempermudah dalam pengambilan atau pembuatan keputusan.



Gamb

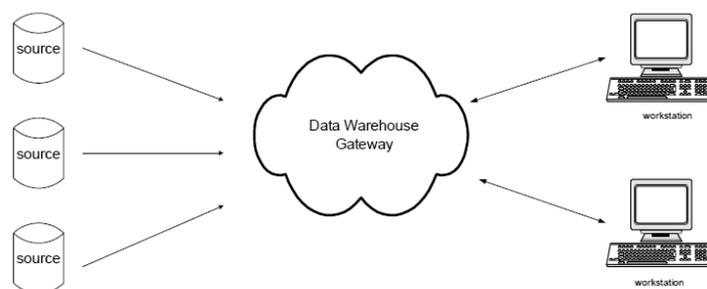
ar 2.5 *Data Warehouse* Terpusat

2.1.2.14.3 *Data Warehouse* Terdistribusi (*Distributed Data Warehouse*)

Adalah sebuah *data warehouse* yang di bentuk dari beberapa buah *data mart* yang masing-masing kemungkinan memiliki skema ETL yang terpisah satu sama lain, atau skema ETL yang sama satu sama lain.

Contoh mudah dari *data warehouse* terdistribusi misalnya adalah sebuah perusahaan besar memiliki sebuah *data warehouse* pusat, dan *data warehouse* pada masing-masing cabang, yang membuat masing-masing cabang dapat dengan mudah menyelesaikan masalah ataupun membuat *policy* yang bisa membantu kemajuan dari cabang tersebut. Hal ini membuat *data mart* yang satu tidak perlu memikirkan apa yang dilakukan *data mart* yang lain.

Namun walaupun memiliki *scope local*, tetapi share informasi tetap bisa dilakukan oleh *data warehouse global* ke *local*, atau dari *local* ke *local* yang lain. Data tersebut tidak di tukar atau disimpan pada *data warehouse* yang lain, dan inilah tugas *data warehouse global* yang bisa menyatukan seluruh *data warehouse local* menjadi satu bagian yang utuh.



Gambar 2.6 *Data Warehouse* Terdistribusi

2.1.2.15 Metodologi Perancangan *Data Warehouse*

Berdasarkan Kimball seperti yang dikutip oleh Connolly dan Begg(2002, p1083), terdapat 9 tahap metodologi dalam pembangunan *data warehouse* yang dikenal dengan *nine-step methodology* yakni:

2.1.2.15.1 Memilih Proses (*Choosing the process*)

Proses(fungsi) merujuk pada subyek masalah dari *data mart* tertentu. *Data mart* yang akan di bangun harus

sesuai anggaran dan dapat menjawab masalah-masalah bisnis yang penting.

2.1.2.15.2 Memilih Grain (*Choosing the grain*)

Memilih *grain* berarti menentukan hal yang sebenarnya dihadirkan oleh tabel fakta. Setelah menentukan *grain-grain* tabel fakta, dimensi-dimensi untuk setiap fakta dapat diidentifikasi.

2.1.2.15.3 Identifikasi dan Membuat Dimensi yang Sesuai (*Identifying and conforming the dimensions*)

Mengidentifikasi dimensi disertai deskripsi detail yang secukupnya. Ketika tabel dimensi berada pada dua atau lebih *data mart*, maka tabel dimensi tersebut harus mempunyai dimensi yang sama atau salah satu merupakan subset dari yang lainnya. Jika suatu tabel dimensi digunakan oleh lebih dari satu *data mart*, maka dimensinya harus disesuaikan.

2.1.2.15.4 Memilih Fakta (*Choosing the facts*)

Memilih fakta yang akan digunakan dalam *data mart*. Semua fakta harus ditampilkan pada tingkat yang

diterapkan oleh *grain* dan fakta juga harus *numeric* dan aditif.

2.1.2.15.5 Menyimpan Pre-kalkulasi di dalam Tabel Fakta (*Storing pre-calculation in the fact tabel*)

Ketika fakta telah dipilih, maka setiap fakta tersebut harus di uji apakah ada fakta yang dapat menggunakan pre-kalkulasi, setelah itu lakukan penyimpanan pada tabel fakta.

2.1.2.15.6 Melengkapi Tabel Dimensi (*Rounding out the dimension tabels*)

Menambahkan sebanyak mungkin deskripsi teks pada tabel dimensi. Deskripsi tersebut harus intuitif dan dapat dimengerti oleh *user*.

2.1.2.15.7 Memilih Durasi dari *Database* (*Choosing the duration of the database*)

Menentukan batas waktu dari umur data yang diambil dan akan dipindahkan ke dalam tabel fakta. Misalnya, data perusahaan dua tahun lalu atau lebih diambil dan dimasukkan dalam tabel fakta.

2.1.2.15.8 Melacak Perubahan dari Dimensi Secara Perlahan (*Tracking slowly changing dimensions*)

Perubahan dimensi yang lambat menjadi sebuah masalah.

Ada 3 tipe dasar dari perubahan dimensi yang lambat, yakni:

- a. Perubahan atribut dimensi yang ditulis ulang (*overwrite*).
- b. Perubahan atribut dimensi yang mengakibatkan pembuatan suatu *record* dimensi yang baru.
- c. Perubahan atribut dimensi yang mengakibatkan sebuah atribut alternatif dibuat, sehingga kedua atribut tersebut yakni atribut yang lama dan atribut yang baru dapat di akses secara bersamaan dalam sebuah dimensi yang sama.

2.1.2.15.9 Memutuskan Prioritas dan Cara Query *(Deciding query priorities and the query modes)*

Mempertimbangkan pengaruh dari perancangan fisik yang akan mempengaruhi persepsi *user* terhadap *data mart*. Selain itu, perancangan fisik akan mempengaruhi masalah administrasi, *backup*, kinerja pengindeksan dan keamanan.

2.1.2.16 Critical Success Factor

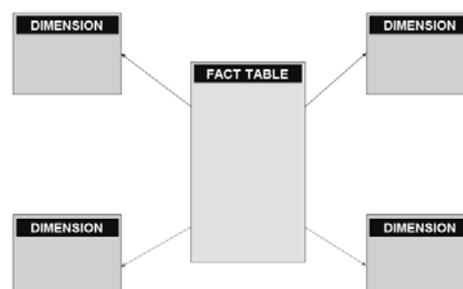
Critical Success Factor merupakan faktor – faktor kritis yang menyebabkan suatu hal bisa dikatakan telah sukses, dan keuntungan-keuntungan apa saja yang didapatkan dengan

kesuksesan tersebut. Dalam pembahasan ini kami ingin membuat sebuah *data warehouse* yang dapat menghasilkan informasi secara akurat sehingga dapat membantu pemimpin untuk mengambil keputusan dengan data hasil dari *data warehouse* dan juga mempermudah seluruh pengaksesan *report* dengan *interface* yang sederhana dan mudah dimengerti (*user friendly*).

2.1.2.17 Konsep Pemodelan *Data Warehouse*

2.1.2.17.1 Skema Bintang (*Star schema*)

Star schema adalah model yang paling mudah di dalam melakukan pemodelan terhadap *data warehouse*. Model ini juga dikenal dengan nama *star join schema*. Pada model ini setiap tabel dimensi hanya berhubungan dengan tabel fakta.

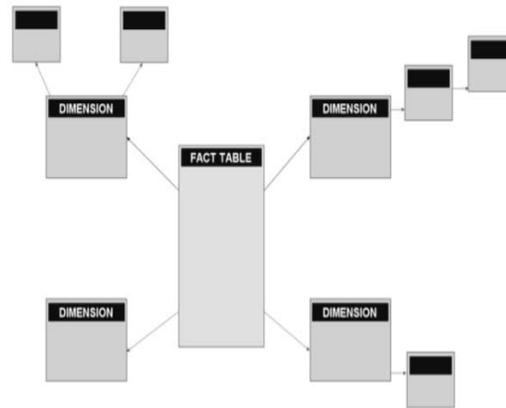


Gambar 2.7 Contoh *Star Schema*

2.1.2.17.2 Skema Snowflake (*Snowflake Schema*)

Pengaturan logical terhadap tabel pada multidimensional *database* atau *data warehouse*, yang di mana dimensinya di

buat lebih mendetail sehingga bentuknya menyerupai bentuk *snowflake*. Secara tidak langsung model ini memiliki hubungan yang erat dengan *star schema*, karena dasar daripada model ini adalah *star schema* itu sendiri.



Gambar 2.8 Contoh *Snowflake Schema*

2.1.2.18 ETL(*Extract, Transform, Load*)

ETL menurut S. Prabhu dan N. Venatesan (2007, p112) adalah proses dimana kita melakukan migrasi dari *database* operasional menuju *data warehouse*. Selain ETL proses ini juga dikenal dengan nama ETT (*Extract, Transformation, Transportation*), atau ETM (*Extract, Transformation, Move*). Proses ETL sendiri di bagi menjadi 3 yaitu:

2.1.2.18.1 *Extract*

Extract adalah proses penentuan *source* yang akan digunakan sebagai sumber data bagi *data warehouse*. Di sini kita bisa menentukan data apa saja yang diperlukan, tabel apa yang dijadikan sumber, dll.

2.1.2.18.2 Transform

Setelah *source* di tentukan, lalu data-data tersebut dirubah agar sesuai dengan *standard* yang ada pada *data warehouse*.

2.1.2.18.3 Load

Sedangkan *Load* adalah proses untuk memasukan data-data yang sudah di *Transformasi* ke dalam *data warehouse* untuk disimpan sebagai *summary* atau *archive*.

2.1.3 Pengantar Berbasis Web

2.1.3.1 Pengertian Berbasis

Menurut Kamus Besar Bahasa Indonesia berbasis merupakan keadaan dimana kita menjadikan sesuatu sebagai basis, yang dalam hal ini basis merupakan sebuah aplikasi yang dapat kita gunakan sebagai *user interface*.

2.1.3.2 Pengertian Web

2.1.3.2.1 Pengertian *Internet*

Internet seperti yang sebutkan oleh Turban, adalah sebuah sistem *global* dari komputer yang saling terhubung, yang menggunakan port *standard* (TCP/IP) yang melayani milyaran *user*. *Internet* adalah kumpulan jaringan yang terdiri dari computer akademik, pemerintah, layanan publik, dll. Umumnya *internet* mengandung banyak sekali sumber informasi dan pelayanan terhadap data, hal yang paling umum diketahui adalah *hypertext* atau dokumen dari WWW serta infrastruktur pendukung *e-mail*. Namun dalam perkembangannya informasi yang disajikan semakin beragam, misalnya *online chat*, *file transfer*, *file sharing*, *gaming*, *commerce*, *social networking*, *publishing*, *video on demand*, *teleconferencing*, dan *telecommunication*.

Pada *Internet* data dikirimkan melalui *packet switching* di dalam *Internet Protocol* atau sering disebut juga dengan IP. IP sendiri diatur menggunakan *set-of-rule* yang digunakan untuk mengirimkan atau pun menerima sebuah paket dari satu mesin, atau mesin lainnya melalui *Internet*. Turban menjelaskan bahwa data adalah bentuk kasar dari sebuah fakta yang digunakan untuk menjelaskan sesuatu, *event*, aktivitas, dan transaksi yang direkam, ditangkap, disimpan, dan disembunyikan, tetapi tidak diatur untuk menyampaikan arti tertentu. Data dikirimkan melalui *packet switching* dimana data/informasi dikelompokkan menjadi paket-paket kecil dan dikirim berdasarkan sebuah jalur yang ditentukan pada *network*. *Internet infrastructure* sendiri disediakan oleh *Internet Service Providers*. Menurut Turban,

Internet Service Providers (ISP) adalah sebuah perusahaan pribadi yang menyediakan servis jaringan dan telekomunikasi jarak jauh yang terhubung dengan sebuah *Internet nodes*.

2.1.3.2.2 Pengertian *World Wide Web* (WWW)

Menurut Turban, WWW adalah bagian dari *Internet* yang berfungsi untuk melakukan fungsi pengiriman pada *Internet*, menggunakan *client/server architecture*, untuk *handle* semua *type* informasi digital seperti, *text*, *hypermedia*, *graphics*, *sounds*. *Web* sendiri adalah sistem yang secara umum menyetujui standarisasi untuk menyimpan, menerima, memformat, dan menampilkan informasi seluruh *type* informasi digital.

2.1.3.2.3 Pengertian *Hyper Text Markup Language* (HTML)

Menurut opini Turban, HTML adalah bahasa pemrograman yang digunakan untuk membuat dan menampilkan sebuah dokumen di *World Wide Web* (WWW). *User* mampu mengontrol elemen visual tanpa mengganti informasi sebenarnya dan mereka bisa melakukan referensi terhadap dokumen yang lain menggunakan *dynamic hyper links*. *Hypertext* adalah pendekatan terhadap *data management* yang menyediakan informasi dimana data tersebut disimpan di dalam *network nodes* yang diakses melalui browser (*interactive browsing*).

2.1.3.2.4 Pengertian *Web Application*

Web application menurut Pressman, adalah sebuah kesatuan files *hypertext* yang saling berkaitan satu sama lain yang digunakan untuk menampilkan informasi menggunakan *text* atau *graphic*, namun sekarang sudah mulai berkembang menjadi sebuah lingkungan komputasi yang menarik yang menyediakan *stand alone features*, fungsi komputasi, penyediaan content bagi *user*, hingga integrasi dalam *database* dan aplikasi bisnis.

Dalam menampilkan *content web application* menggunakan *web page* sebagai sarana. Menurut Powell sendiri, *Web Page* adalah dokumen yang bisa diterima melalui *network* dan ditampilkan melalui *browser* berdasarkan *request* yang diinginkan oleh *user*. Dokumen ini mungkin saja terdiri dari *text*, *images*, atau video.

2.1.3.2.5 Pengertian Web Server

Whitten menyatakan bahwa, *Web server* adalah sebuah *computer server* yang menghost *internet* atau *intranet*. Salah satu contoh *web server* yang sangat populer adalah Apache, karena ia dapat berjalan pada berbagai macam platform dan sistem operasi, seperti yang dijelaskan oleh Narnmore. *Modular designnya* memiliki berbagai macam jenis servis dan salah satu servis yang disediakan oleh Apache adalah *Server-Side Scripting*.

2.1.3.2.6 Pengertian PHP

Whitten menjelaskan PHP sebagai sebuah bahasa pemrograman sederhana yang didesain khusus untuk aplikasi *internet*. Salah satu keuntungan dalam melakukan implementasi menggunakan *server-side scripting* adalah balasan kepada *user (client)* dapat di kostumnisasi berdasarkan permintaan yang mereka inginkan, hak akses, atau *query* ke tempat penyimpanan data.

Sedangkan sebuah *server-side scripting* yang membuat *web site* anda menjadi benar-benar dinamis. Code dari PHP sendiri disisipkan kedalam kode HTML dan akan dijalankan ketika *web* tersebut dikunjungi. Karena PHP adalah *server-side scripting*, maka script PHP akan dijalankan pada *server side*, bukan *client side*, hal ini menyebabkan *web site* kita menjadi dinamis.

Salah satu keuntungan PHP adalah *cross-platform* yang berarti PHP dapat digunakan pada sistem operasi berbasis Windows, UNIX, Java ataupun MAC. Oleh karena itu PHP banyak digunakan dalam melakukan *development* terhadap sebuah *website*.

Dibandingkan dengan bahasa *server-side* yang lain, PHP memiliki keunggulan berupa:

- a. *High-performance* dan sangat efisien, karena dapat diimplementasikan pada *server* dengan *requirement* yang rendah dan tidak *menuntut resource* yang besar. Selain itu PHP akan berjalan sama baiknya, baik pada *server* yang

memiliki *resource* yang rendah maupun *server* yang memiliki *resource* yang besar.

- b. Integrasi terhadap *database*. Ini berarti PHP memiliki *interfaces* untuk berhubungan dengan berbagai macam *database* sistem. Ada banyak sekali *native connection* yang bisa digunakan untuk melakukan koneksi ke *database* sistem seperti : MySQL, MSSQL, PostgreSQL, Oracle, dll. PHP juga bisa menggunakan ODBC untuk melakukan koneksi ke dalam *database* sistem. Selain itu PHP juga mengembangkan PDO untuk memperkaya fitur *database* mereka, dimana *database* akan dianggap sebagai sebuah *object* yang tentu saja membuatnya mudah untuk dimanipulasi.
- c. PHP sangat murah, karena PHP disebarakan secara gratis di *internet*.
- d. Sangat mudah untuk mempelajari bahasa PHP, terutama untuk orang-orang yang sudah pernah belajar bahasa pemrograman lainnya, karena pada dasarnya *syntax* PHP menggunakan konsep dari C dan Perl.
- e. PHP sangat portabel, yang berarti PHP bisa berjalan pada berbagai macam sistem operasi. Bahkan *code* PHP sendiri tidak perlu dirubah untuk berjalan pada sistem operasi yang berbeda.

- f. *Source code* dari PHP sendiri tersedia secara gratis, sehingga kita dapat dengan mudah melakukan *tweaking* dan mencari *bug*, sehingga tidak perlu *menunggu patch* resmi dari *developer*.

2.1.3.3 Pengertian Berbasis Web

Jadi pengertian Berbasis Web adalah penggunaan *web* sebagai penyalur (*interface*) antara *user* dengan *data warehouse*. Di mana kita dapat menggunakan program-program seperti Adobe Dreamweaver untuk membuat *website* dengan berbagai macam bahasa pemrograman.

System berbasis web diciri-cirikan sebagai berikut:

1. Umumnya berbasis pada jaringan atau network yang menghubungkan 1 komputer dan computer lain.
2. Menggunakan Aplikasi Web Server untuk menanggapi request dari client ke server.
3. Menggunakan protocol HTTP dalam transfer data.
4. Tidak bergantung pada sebuah System Operasi tertentu.
5. Menggunakan Web Browser untuk mempresentasikan data yang dikirim.

2.1.4 Hubungan Data Warehouse dengan Berbasis Web

Data warehouse digunakan untuk menampung summary yang didapat dari OLTP. Data yang disimpan harus dipresentasikan untuk bisa di analisa

lebih lanjut oleh user yang membutuhkan, dalam hal ini Menteri atau Pejabat Esselon 1.

Salah satu cara mempresentasikan data adalah dengan menggunakan Web, karena Web dapat mengirimkan informasi media, seperti text, gambar, tabel secara cepat dan mudah.

Sehingga *Data warehouse* yang berbasis web pada dasarnya adalah sebuah *Data warehouse* yang mempresentasikan datanya menggunakan Web sebagai basisnya.

2.2 Teori-teori Khusus

2.2.1 Pengelolaan Energi

Menurut Kamus Besar Bahasa Indonesia, pengelolaan berarti proses yang memberikan pengawasan kepada semua hal yang terlibat dalam pelaksanaan kebijaksanaan dan pencapaian tujuan seperti produksi, konsumsi, impor dan ekspor. Sedangkan energi berarti daya (kekuatan) yang dapat digunakan untuk melakukan berbagai proses kegiatan. Jadi pengelolaan energi adalah suatu proses pengawasan daya (kekuatan) untuk menjamin agar proses itu berjalan dengan lancar sesuai dengan tugasnya dan jika mengalami kendala maka masalah tersebut dapat diselesaikan dengan segera.

Dalam pembahasan ini Kementerian Energi dan Sumber Daya Mineral membagi energi menjadi 2 jenis, yaitu:

2.2.1.1 Energi Primer

Energi Primer merupakan energi yang belum diproses, energi primer terdiri dari minyak bumi, gas bumi, batu bara, panas bumi, tenaga air, dan biomasa. Yang dalam skripsi ini kami akan memproses:

- a. Minyak Bumi : Produksi, Konsumsi, Ekspor dan Impor.
- b. Gas Bumi : Produksi dan Konsumsi.
- c. Panas Bumi : Produksi.

2.2.1.2 Energi Final

Energi final merupakan energi yang telah diolah sampai ketitik akhirnya, energi final terdiri dari BBM, Non BBM, LNG, dan LPG. Yang dalam skripsi ini kami akan memproses:

- BBM : Konsumsi, Produksi, Ekspor dan Impor.
- Non BBM : Produksi, Ekspor dan Impor.
- LNG : Ekspor dan Impor.
- LPG : Ekspor dan Impor.